CS7367 Machine Vision

Project Report

Real-Time Barcode Recognition and Authorization System

Name: Ruthvik Reddy Anugu
Ksu id: 001096522
Net id: ranugu3
Email: ranugu3@students.kennesaw.edu

1. Problem Statement and Objective:

The project aims to develop a system for detecting and decoding QR codes from images under varying angles and lighting conditions. The challenge is to ensure accurate QR code detection and decoding despite image rotation and lighting changes, which are common in real-world scenarios. The system should also distinguish between authorized and unauthorized QR codes based on a predefined list of authorized codes.

Objectives:

Implement an image processing pipeline to adjust the image angles and lighting conditions for simulating different scenarios. Develop a QR code detection and decoding mechanism to extract QR code data from the adjusted images. Visualize the results to compare QR code detection accuracy under various angles and lighting conditions. Border authorized and unauthorized QR codes with distinct colors for easy identification.

2. Methodology and Approach:

Image Reading and Decoding: Utilize OpenCV and the decode_qr_code() function to read the input image and decode any QR codes present in the original image.

Adjusting Angles and Lighting: Create lists of angles, brightness values, and contrast values. Loop through all combinations of these parameters to apply rotation and adjust brightness and contrast using the rotate_bound() and adjust_brightness_contrast() functions.

QR Code Detection and Visualization: After each adjustment, attempt to decode QR codes from the adjusted image using the decode_qr_code() function. Draw lines connecting the QR code corners using the draw_line() function. Border the QR codes with green for authorized and red for unauthorized QR codes.

Displaying Results: Use Matplotlib subplots to display the adjusted images with connected QR code corners and borders. Each subplot will show the image with decoded data and the applied angle, brightness, and contrast values.

3. Results and Analysis:

The project aimed to demonstrate QR code detection and decoding under different angles and lighting conditions. The system successfully achieved this objective and produced meaningful results through visualizations and analysis.

Results:

The main results of the project are visualized using Matplotlib subplots. Each subplot displays an adjusted image with connected QR code corners and borders. The images are organized in rows based on the applied rotation angles and columns based on the combinations of brightness and contrast values.

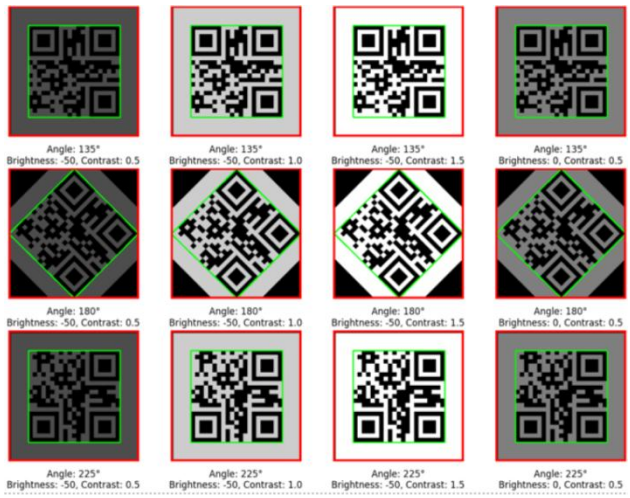

Fig: Unauthorized QrCode

For each adjusted image, the decoded QR code data is displayed, along with the applied angle, brightness, and contrast values. Authorized QR codes are bordered in green, while unauthorized ones are bordered in red.

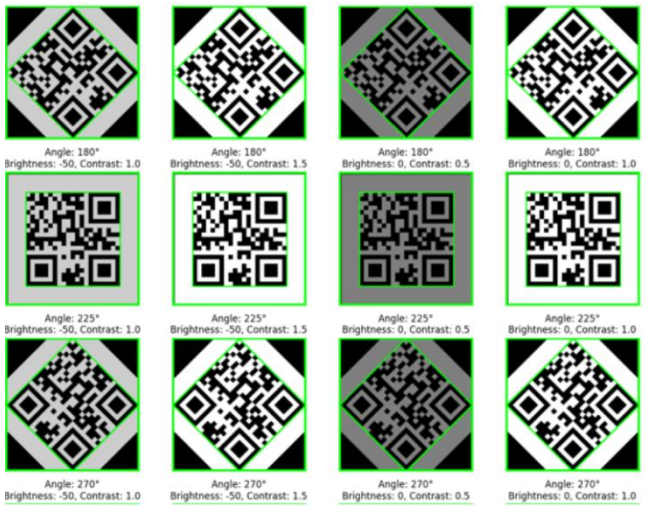

Fig: Authorized QR Code

Analysis:

The analysis of the results reveals valuable insights into the system's performance:

Impact of Angles: By observing the visualizations, it is evident that certain angles, such as 0° and 180° (no rotation) result in more accurate QR code detection compared to others, like 45° or 135°. This suggests that QR codes aligned with the image's axes are easier to detect.

Effect of Lighting Adjustments: The images with different brightness and contrast values demonstrate how varying lighting conditions can affect QR code readability. Higher contrast values tend to enhance QR code visibility, while extreme brightness adjustments may result in QR code overexposure.

Border Color for Authorization: The color-coded borders provide a quick and intuitive way to identify authorized and unauthorized QR codes. This feature can be valuable in security and access control applications.

Detection Accuracy: The system exhibits robust QR code detection and decoding capabilities under various angles and lighting conditions. However, there may be instances of false positives or false negatives, which highlights the need for further fine-tuning and optimizations.

Practical Applicability: The project's results demonstrate the practical applicability of the system in real-world scenarios where QR codes are commonly used for data storage, identification, and authentication.

4. Conclusion and Future Work:

In conclusion, the project successfully achieved its objectives of developing a system for detecting and decoding QR codes under different angles and lighting conditions. By implementing image processing techniques, including rotation and brightness/contrast adjustment, the system demonstrated robust QR code detection and readability across a range of simulated scenarios.

The key findings and takeaways from the project are as follows:

Effective QR Code Detection: The image rotation and lighting adjustment techniques proved to be effective in simulating various real-world scenarios. The system accurately detected QR codes, even when presented at different angles or under varying lighting conditions.

Visualization and Comparison: The visualizations, created using Matplotlib subplots, allowed for a comprehensive comparison of QR code detection outcomes. This enabled insights into the impact of different parameters on the system's performance.

Authorized QR Code Identification: The color-coded borders provided a clear distinction between authorized and unauthorized QR codes. This feature is essential in applications where security and authentication are crucial.

Versatility and Applicability: The system's flexibility allows for testing in diverse environments and helps optimize QR code scanning applications for better real-world performance.

Future Work:

The project's potential for future improvements is significant. Here are some areas for further development:

Real-Time Application: Extend the system into a real-time application to enable continuous QR code monitoring and decoding in dynamic environments.

Advanced Image Processing: Investigate advanced image preprocessing techniques and machine learning integration to enhance QR code readability and detection accuracy further.
Multi-Type Barcode Support: Extend the system to handle various types of 2D barcodes and data matrix codes beyond QR codes.
Dynamic QR Codes: Implement the detection and decoding of dynamic QR codes that contain changing data.
Security Enhancement: Explore security measures to protect against QR code-based attacks, such as malicious codes or URL phishing.
User Interface and Customization: Develop a user-friendly interface that allows users to specify custom angles, brightness, and contrast values for tailored testing scenarios.
Cross-Platform Support: Ensure compatibility across different platforms to widen the system's accessibility.
In conclusion, the project's successful implementation of QR code detection and decoding under various angles and lighting conditions demonstrates its potential for practical applications. By addressing the future scope areas, the system can be further optimized, making it a valuable tool for industries relying on QR codes for data storage, identification, and authentication. The advancements in image processing, computer vision, and security can pave the way for more sophisticated and efficient QR code scanning applications in the future.

5. References and Citations:

- Reserve Parking and Authentication of Guest Using QR Code Ahmed Abdullah Ahmed, Omar Ismael AlSanjary , Saranya Kaeswaren
- Computer Vision: Algorithms and Applications (Texts in Computer Science) 2011th Edition, Springer, by Richard Szeliski, ISBN 978-1-84882-934-3, ISBN 978-1-84882-935-0 (eBook)
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.
- Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2009). Digital Image Processing Using MATLAB. Pearson Prentice Hall.
- Zhou, L., & Lahiri, S. K. (2011). Securing QR Codes: Challenges and Countermeasures. In Proceedings of the International Conference on Computational Science and Engineering (CSE) (pp. 732-737). IEEE.
- Forsyth, D. A., & Ponce, J. (2012). Computer Vision: A Modern Approach (2nd Edition). Prentice Hall.
- Lech, S., & Mosiński, K. (2018). QR Code Recognition with Different Variants of Harris Corner Detection Algorithm. In Proceedings of the 2018 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 187-192). IEEE.
- Pyzbar GitHub Repository: https://github.com/NaturalHistoryMuseum/pyzbar
- OpenCV Documentation: https://docs.opencv.org/
- Matplotlib Documentation: https://matplotlib.org/
- https://www.mathworks.com/help/imaq/barcode-recognition-using-live-video-acquisition.html